

Commutative Languages and their Composition by Consensual Methods^{*}

Stefano Crespi Reghizzi
Pierluigi San Pietro

DEIB, Politecnico di Milano and CNR-IEIT

stefano.crespireghizzi@polimi.it pierluigi.sanpietro@polimi.it

Commutative languages with the semilinear property (SLIP) can be naturally recognized by real-time NLOG-SPACE multi-counter machines. We show that unions and concatenations of such languages can be similarly recognized, relying on – and further developing, our recent results on the family of consensually regular (CREG) languages. A CREG language is defined by a regular language on the alphabet that includes the terminal alphabet and its marked copy. New conditions, for ensuring that the union or concatenation of CREG languages is closed, are presented and applied to the commutative SLIP languages. The paper contributes to the knowledge of the CREG family, and introduces novel techniques for language composition, based on arithmetic congruences that act as language signatures. Open problems are listed.

1 Introduction

This paper focuses on commutative languages having the semilinear property (SLIP). We recall that a language has the *linear property* (LIP) if, in any word, the number of letter occurrences (also named Parikh image) satisfies a linear equation; it has the *semilinear property* (SLIP) [5] if the number satisfies one out of finitely many linear equations. A language is *commutative* (COM) if, for every word, all permutations are in the language; thus, the legality of a word is based only on the Parikh image, not on the positions of the letters. Here we deal with the subclass of COM languages enjoying the SLIP, denoted by COM-SLIP, for which we recall some known properties. For a binary alphabet, COM-SLIP languages are context-free whereas, in the general case, they can be recognized by *multi-counter machines* (MCM), in particular by non-deterministic quasi-real-time *blind* MCM (equivalent to *reversal-bounded* MCM [7]). The COM-SLIP family is closed under all Boolean operations, homomorphism and inverse homomorphism, but it is not closed under concatenation.

Our contribution is to relate two seemingly disparate language families: on one hand, the COM-SLIP languages and their closure under union and concatenation (denoted by $\text{COM-SLIP}^{\cup, \cdot}$), on the other hand, the family of *consensually regular* languages (CREG), recently introduced by the authors, to be later presented. We briefly explain the intuition behind it. Given a terminal alphabet, a CREG language is specified by means of a regular language (the *base*) having a *double* alphabet: the original one and a *dotted* copy. Two or more words in the base language *match*, if they are all identical when the dots are disregarded and, in every position, exactly one word has an undotted letter (thus in all remaining words the same position is dotted). In our metaphor, we say that, position by position, one of the base words “places” a letter and the remaining words “consent” to it. A word is in the consensual language if the base language contains a set of matching words, identical to the given word when the

^{*}Work partially supported by PRIN 2010LYA9RH-006 “Automi e linguaggi formali: Aspetti Matematici e Applicativi”.

[†]The main results have been announced in [15], with preliminary sketchy proofs entirely superseded by the present ones.

dots are disregarded. This mechanism somewhat resembles the model of alternating non-deterministic finite automata, but the criterion by which the parallel computations match is more flexible and produces a recognition device which is a MCM working in NLOG-SPACE. This MCM can be viewed as a token or multi-set machine; it has one counter for each state of the DFA recognizing the base language; each counter value counts the number of parallel threads that are currently active in each state. Our main result is that the $\text{COM-SLIP}^{\sqcup, \cdot}$ family is strictly included in CREG; we also prove some non-closure properties of $\text{COM-SLIP}^{\sqcup, \cdot}$.

To construct the regular language that serves as base for the consensual definition of a $\text{COM-SLIP}^{\sqcup, \cdot}$ language, we have devised a new method, which may be also useful to study the inclusion in consensual classes of other families closed union or concatenation. It is easy to consensually specify a COM-LIP language by means of a regular base; however, in general, union or concatenation of two regular bases consensually specifies a larger language than the union or concatenation of the components. To prevent this to happen, we assign a distinct numeric congruence class to each base, which determines the positions where a letter may be placed as dotted or as undotted. For a given word, such positions are not the letter orders, but they are the orders of the letters in the projections of the word on each letter of the alphabet. The congruence acts as a sort of signature that cannot be mismatched with other signatures.

To hint to a potential application, $\text{COM-SLIP}^{\sqcup, \cdot}$ offers a rather suitable schema for certain parallel computation systems, such as Valiant's "bulk synchronous parallel computer" [16]. There, when all threads in a parallel computational phase, which we suggest to model by a commutative language, terminate, the next phase can start; the sequential composition of such phases can be represented by language concatenation; and the composition of alternative subsystems can be modeled by language union. As said, such computation schema is not finite-state but it is a MCM.

Paper organization: Sect. 2 contains preliminaries, some simple properties of $\text{COM-SLIP}^{\sqcup, \cdot}$ and the consensual model. Sect. 3 introduces the decomposed form, states and proves the conditions that ensure union- and concatenation-closure, and details the congruence based constructions. Sect. 4 proves the main result through a series of lemmas. The last section refers to related work and mentions some unanswered questions.

2 Preliminary Definitions and Properties

The terminal alphabet is denoted by $\Sigma = \{a_1, \dots, a_k\}$, the empty word by ε and $|x|$ is the length of a word x . The projection of x on $\Delta \subseteq \Sigma$ is denoted by $\pi_\Delta(x)$; $|x|_a$ is shorthand for $|\pi_{\{a\}}(x)|$ for $a \in \Sigma$, and $|x|_\Delta$ stands for $|\pi_\Delta(x)|$. The i -th letter of x is $x(i)$ and $x(i, j)$ is the substring $x(i) \dots x(j)$, $1 \leq i \leq j \leq |x|$. The *shuffle* operation is denoted by \sqcup .

The *Parikh image* or *vector* of a word $x \in \Sigma^*$ is $\Psi(x) = [|x|_{a_1}, \dots, |x|_{a_k}]$; it can be naturally extended to a language. The component-wise addition of two vectors is denoted by $\vec{p}' + \vec{p}''$. The *commutative closure* of $L \in \Sigma^*$ is $\text{com}(L) = \{x \in \Sigma^* \mid \Psi(x) \in \Psi(L)\}$. A language L is *commutative* if $\text{com}(L) = L$; the corresponding language family is named COM. A language $L \subseteq \Sigma^*$ has the *linear property* (LIP) if there exist $q+1 > 0$ vectors $\vec{c}, \vec{p}^{(1)}, \dots, \vec{p}^{(q)}$ over \mathbb{N}^k , (resp. the *constant* and the *periods*) such that $\Psi(L) = \{\vec{c} + n_1 \cdot \vec{p}^{(1)} + \dots + n_q \cdot \vec{p}^{(q)} \mid n_1, \dots, n_q \geq 0\}$.

A language has the *semilinear property* (SLIP) if it is the finite union of LIP languages. The families of commutative LIP/SLIP languages are denoted by *COM-LIP*/ *COM-SLIP*, respectively. It is well known that COM-SLIP is closed under the Boolean operations, inverse homomorphism, homomorphism and Kleene star, but not under concatenation, which in general destroys commutativity. However, the concatenation of COM-SLIP languages still enjoys the SLIP.

Let $\text{COM-SLIP}^{\cup, \cdot}$ be the smallest family including COM-SLIP languages and closed under union and concatenation. Let BLIND denote the class of languages accepted by nondeterministic, blind multi-counter machines [7], which, we recall, are restricted to perform a test for zero only at the end of a computation; they are equivalent to reversal-bounded counter machines. The following facts, although to our knowledge not stated in the literature, are straightforward.

Proposition 1. Main Properties of $\text{COM-SLIP}^{\cup, \cdot}$.

1. Every $\text{COM-SLIP}^{\cup, \cdot}$ language on a binary alphabet is context-free.
2. $\text{COM-SLIP}^{\cup, \cdot} \subsetneq \text{BLIND}$.
3. The $\text{COM-SLIP}^{\cup, \cdot}$ family is not closed under intersection and Kleene star.

Proof. Let $L' = \text{com}((ab)^+)$. Statement (1) is immediate: since all COM-SLIP on a binary alphabet are context-free [9, 13], also their union and concatenation is context-free. Statement (2) is also immediate, since COM-SLIP is clearly included in BLIND, and BLIND is closed by union and concatenation. The inclusion is strict since BLIND includes also non-context-free languages on a binary alphabet [7]. To prove non-closure of intersection – Statement (3) – assume by contradiction that the language $L_0 = L' \cap a^+b^+ = \{a^n b^n \mid n > 0\}$ is in $\text{COM-SLIP}^{\cup, \cdot}$. Hence, also the languages $L_1 = \{a^+ b^n a^n \mid n > 0\}$, $L_2 = \{a^m b^m a^+ \mid m > 0\}$ and $L_1 \cap L_2 = \{a^n b^n a^n \mid n > 0\}$ are in $\text{COM-SLIP}^{\cup, \cdot}$. But the latter language is not context-free, contradicting Statement (1). To complete the proof of Statement (3), if $\text{COM-SLIP}^{\cup, \cdot}$ were closed under Kleene star, then language $L_3 = (L'c)^*$ would be $\text{COM-SLIP}^{\cup, \cdot}$, with $c \notin \{a, b\}$. However, $\text{COM-SLIP}^{\cup, \cdot}$ is included in BLIND, which is an intersection-closed full semiAFL (see Section 5 of [1] and also Theorem 1 of [7]), i.e., BLIND is closed under intersection, union, arbitrary homomorphism, inverse homomorphism, and intersection with regular languages. Hence, the language $L_4 = L_3 \cap (a^+b^+c)^* = \{a^n b^n c \mid n > 0\}^*$ would be in BLIND. Letter c can be deleted by a homomorphism, hence also the language $\{a^n b^n \mid n > 0\}^*$, is BLIND, contradicting Corollary 3 of [1] and also Theorem 6, Part (2), of [7]. \square

2.1 Consensual Languages.

We present the necessary elements of consensual language theory [2, 3]. Let $\tilde{\Sigma}$ be the *dotted* (or marked) copy of alphabet Σ . For each $a \in \Sigma$, \tilde{a} denotes the set $\{a, \tilde{a}\}$. The alphabet $\tilde{\Sigma} = \Sigma \cup \tilde{\Sigma}$ is named *double* (or *internal*). To express a sort of agreement between words over the double alphabet, we introduce a binary relation, called *match*, over $\tilde{\Sigma}^*$.

Definition 1 (Match). The partial, symmetrical, and associative binary operator, called *match*, $@ : \tilde{\Sigma} \times \tilde{\Sigma} \rightarrow \tilde{\Sigma}$ is defined as follows, for all $a \in \Sigma$:

$$\begin{cases} a@ \tilde{a} = \tilde{a}@a = a \\ \tilde{a}@ \tilde{a} = \tilde{a} \\ \text{undefined} & \text{in every other case.} \end{cases}$$

The match is naturally extended to strings of equal length, as a letter-by-letter application, by assuming $\varepsilon@ \varepsilon = \varepsilon$: for every $n > 1$, for all $w, w' \in \tilde{\Sigma}^n$, if $w(i)@w'(i)$ is defined for every $i, 1 \leq i \leq n$, then

$$w @ w' = (w(1)@w'(1)) \cdot \dots \cdot (w(n)@w'(n)). \quad \text{In every other case, } w@w' \text{ is undefined.}$$

Hence, the match is undefined on strings w, w' of unequal lengths, or else if there exists a position j such that $w(j)@w'(j)$ is undefined, which occurs in three cases: when both characters are in Σ , when both are in $\tilde{\Sigma}$ and differ, and when either one is dotted but is not the dotted copy of the other. Syntactically, the precedence of the match operator is just under the precedence of the concatenation. The match w of two or more strings is further qualified as *strong* if $w \in \Sigma^*$, or as *weak* otherwise. By Def. 1, if $w = w_1@w_2@ \dots @w_m$ is a strong match of $m \geq 1$ words w_1, \dots, w_m , then in each position $1 \leq i \leq |w|$, exactly one word, say w_h , is undotted, i.e., $w_h(i) \in \Sigma$, and $w_j(i) \in \tilde{\Sigma}$ for all $j \neq h$; we say that word w_h *places* the letter at position i and the other words *consent* to it. Metaphorically, the words that strongly match provide mutual consensus on the validity of the corresponding word over Σ , thereby motivating the name “consensual” of the language family.

The match is extended to two languages B', B'' on the double alphabet, as $B' @ B'' = \{w' @ w'' \mid w' \in B', w'' \in B''\}$. The iterated match $B^{i@}$ is defined for all $i \geq 0$, as $B^{0@} = B$, $B^{i@} = B^{(i-1)@} @ B$, if $i > 0$.

Definition 2 (Consensual language). The *closure under match*, or *@-closure*, of a language $B \subseteq \tilde{\Sigma}^*$ is $B^@ = \bigcup_{i \geq 0} B^{i@}$. The *consensual language with base B* is defined as $\mathcal{C}(B) = B^@ \cap \Sigma^*$. The family of *consensually regular* languages, denoted by CREG, is the collection of all languages $\mathcal{C}(B)$, such that the base B is regular.

It follows that a CREG language can be *consensually specified* by a regular expression over $\tilde{\Sigma}$.

Example 1. The LIP language $L = \{a^n b^n c^n \mid n > 0\}$ is consensually specified by the base (that we may call a “consensual regular expression”) $\hat{a}^* \hat{a} \hat{a}^* \hat{b}^* \hat{b} \hat{b}^* \hat{c}^* \hat{c} \hat{c}^*$. For instance, $aabbcc$ is the (strong) match of $\hat{a} \hat{a} \hat{b} \hat{b} \hat{c} \hat{c}$ and $\hat{a} \hat{a} \hat{b} \hat{b} \hat{c} \hat{c}$. The commutative closure of L is also in CREG, with base: $\text{com}(abc) \sqcup \tilde{\Sigma}^*$.

Similarly, the COM-LIP language $L' = \text{com}((ab)^+) = \mathcal{C}(B_1)$, where $B_1 = \text{com}(ab) \sqcup \tilde{\Sigma}^*$.

The COM-LIP language $L'' = \text{com}((abb)^+)$ is specified by the base $B_2 = \text{com}(abb) \sqcup \tilde{\Sigma}^*$.

The languages $L' \cup L''$ and $L' \cdot L''$ are in CREG, but, counter to a naive intuition, they are not specified by the bases obtained by composition, respectively, $B_1 \cup B_2$ and $B_1 B_2$. In general $\mathcal{C}(B_1 \cup B_2) \supset \mathcal{C}(B_1) \cup \mathcal{C}(B_2)$: in the examples, $\mathcal{C}(B_1 \cup B_2)$ contains also undesirable “cross-matching” words, such as $ababb = ab\hat{a}\hat{b}\hat{b} @ \hat{a}babb$. A systematic compositional technique for obtaining the correct bases for the union and concatenation is the main contribution of this paper.

Summary of known and relevant CREG properties. *Language family comparisons:* CREG includes the regular languages, is incomparable with the context-free and deterministic context-free families, is included within the context-sensitive family, and it contains non-SLIP languages. CREG strictly includes the family of languages accepted by partially-blind multi-counter machines that are deterministic and quasi-real-time, as well as their union [4].

Closure properties: CREG is closed under marked concatenation, marked iteration, inverse alphabetic homomorphism, reversal, and intersection and union with regular languages. The marked concatenation of two languages $L_1, L_2 \subseteq \Sigma^*$ is the language $L_1 \# L_2$, where $\# \notin \Sigma$, while the marked iteration of $L \subseteq \Sigma^*$ is the language $(L\#)^*$. A language family enjoying such properties is known as a *pre-Abstract Family of Languages* (see, e.g., [14]). A precise characterization of the bases that consensually specify regular languages is in [3]; an analysis of the reduction in descriptive complexity of the consensual base with respect to the specified regular language is in [2].

Complexity: CREG is in NLOGSPACE, i.e., NSPACE($\log n$) (often called NL): it can be recognized by a nondeterministic multitape Turing machine working in $\log n$ space. The recognizer of CREG languages is a special kind of nondeterministic, real-time multi-counter machine.

Useful notations for consensual languages. The following mappings will be used:

switching $switch : \tilde{\Sigma} \rightarrow \tilde{\Sigma}$ where $switch(a) = \dot{a}$, $switch(\dot{a}) = a$, for all $a \in \Sigma$
 marking $dot : \tilde{\Sigma} \rightarrow \dot{\Sigma}$ where $dot(x) = x$, if $x \in \dot{\Sigma}$, and $dot(x) = \dot{a}$, if $x = a \in \Sigma$
 unmarking $undot : \tilde{\Sigma} \rightarrow \Sigma$ where $undot(a) = switch(dot(a))$, for all $a \in \Sigma$.

These mappings are naturally extended to words and languages, e.g., given $x \in \tilde{\Sigma}^*$, $switch(x)$ is the word obtained interchanging a and \dot{a} in x (a sort of “complement”).

In the remainder of the paper, we assume that each base language is a subset of $\tilde{\Sigma}^* - \dot{\Sigma}^+$, since words in $\dot{\Sigma}^+$ are clearly useless in a match. Let B, B' be languages included in $\tilde{\Sigma}^* - \dot{\Sigma}^+$. We say that B is *unproductive* if $\mathcal{C}(B) = \emptyset$, and that the pair (B, B') is *unmatchable* if $B @ B' = \emptyset$.

3 Consensual specifications composable by union and concatenation

Since it is unknown whether the whole CREG family is closed under union and concatenation, we first introduce a normal form, named decomposed,¹ of the base languages, which is convenient to ensure such closure properties. Second, we state two further conditions, named joinability and concatenability, for decomposed forms, and we prove that they, respectively, guarantee closure under union and concatenation. Such results hold for every consensual language, but the difficulty remains to find a systematic method for constructing base languages that meets such conditions. Third, in Sect. 3.1 we introduce an implementation of decomposed forms, relying on numerical congruences, that will permit us to prove in Sect. 4 that the (\cup, \cdot) -closure of commutative SLIP languages is in CREG.

Definition 3 (Decomposed form). A base $B \subseteq \tilde{\Sigma}^* - \dot{\Sigma}^+$ has the *decomposed form* if there exist a (disjoint) partition of B into two languages, named the *scaffold* sc and the *fill* fl of B , such that fl is unproductive, and the pair (sc, sc) is unmatchable.

The names scaffold and fill are meant to convey the idea of an arrangement superposed just once on each word of the base and, respectively, of an optional (but repeatable) component to complete the letters which are dotted in the scaffold. Three straightforward remarks follow. For every base B there exists a consensually equivalent decomposed base: it suffices to take as scaffold the language $\{a \dot{dot}(y) \mid ay \in B, a \in \Sigma, y \in \tilde{\Sigma}^*\}$, and as fill the language $\{\dot{dot}(x)y \mid x \in \tilde{\Sigma}, y \in \tilde{\Sigma}^*, xy \in B\}$. For every $s \subseteq sc$, $f \subseteq fl$, the base $s \cup f$ is a decomposed form. The scaffold, but not the fill, may include words over Σ .

Consider a word $w \in \mathcal{C}(B)$. Since the fill is unproductive, its match closure cannot *place* all the letters of w and such letters must be placed by the scaffold. Since by definition the match closure of the scaffold alone is the scaffold itself, the following fundamental lemma immediately holds.

Lemma 1. *If $B = sc \cup fl$ is in decomposed form, as in Def. 3, then $\mathcal{C}(B) = (sc \cup (sc @ fl^@)) \cap \Sigma^*$.*

Example 2. The table shows the decomposed bases of languages $com((ab)^+)$ and $com((abb)^+)$ of Sect. 2.1, considering for brevity only the case that the number of a ’s is a multiple of 3. Let $L' = com(\{a^{3n}b^{3n} \mid n \geq 1\})$, with scaffold sc' and fill fl' , and $L'' = com(\{a^{3n}b^{6n} \mid n \geq 1\})$, with scaffold sc'' and fill fl'' :

	scaffold	fill	a strong match
L'	$(a\dot{a}a)^+ \sqcup (b\dot{b}b)^+$	$(\dot{a}^3)^* \dot{a}a\dot{a}(\dot{a}^3)^* \sqcup (\dot{b}^3)^* \dot{b}b\dot{b}(\dot{b}^3)^*$	$\begin{array}{ccccccc} a & b & \dot{a} & a & \dot{b} & b \in sc' \\ @ & \dot{a} & \dot{b} & a & \dot{a} & b & \dot{b} \in fl' \end{array}$
L''	$(\dot{a}a\dot{a})^+ \sqcup (\dot{b}b\dot{b})^+$	$(\dot{a}^3)^* a\dot{a}a(\dot{a}^3)^* \sqcup (\dot{b}^3)^* \dot{b}b\dot{b}(\dot{b}^3)^*$	$\begin{array}{ccccccc} \dot{a} & \dot{b} & a & a & b & b & \dot{b} & b \in sc'' \\ @ & a & b & \dot{a} & \dot{a} & \dot{b} & \dot{b} & \dot{b} \in fl'' \end{array}$

¹In [4], we introduced the idea of a decomposed form for certain multi-counter machines, but that definition does not work for commutative languages.

Clearly, every word in sc' is unmatchable with every other word in sc' , hence $sc'@sc' = \emptyset$. Similarly, every fill is unproductive. Every word in L' is the match of exactly one word in the scaffold with one or more words in the fill. Analogous remarks hold for L'' .

Next, imagine to consensually specify two languages by bases in decomposed form $B' = sc' \cup fl'$ and $B'' = sc'' \cup fl''$. By imposing additional conditions on the bases, we obtain two very useful theorems about composition by union and concatenation.

Definition 4 (Joinability). Two base languages B', B'' in decomposed form are *joinable* if their union $B' \cup B''$ is decomposed, with scaffold $sc' \cup sc''$ and fill $fl' \cup fl''$, and the pairs (sc', fl'') and (sc'', fl') are unmatchable.

Theorem 1 (Union of consensual languages in decomposed form). *Let the base languages B', B'' be in decomposed form. If B' and B'' are joinable then $\mathcal{C}(B') \cup \mathcal{C}(B'') = \mathcal{C}(B' \cup B'')$.*

Proof. It suffices to prove the inclusion $\mathcal{C}(B' \cup B'') \subseteq \mathcal{C}(B') \cup \mathcal{C}(B'')$, since the opposite inclusion is obvious by Def. 2. Let $x \in \mathcal{C}(B)$. Since B is decomposed, by Lemma 1 it must be either $x \in sc@fl@$ or $x \in sc$. In the latter case, x is in B' or in B'' , and the inclusion follows. In the former case, there exist $n \geq 2$ words w_1, w_2, \dots, w_n , with $n \leq |x|$, $w_1 \in sc$, $w_2, \dots, w_n \in fl$ and $w_1@w_2@ \dots @w_n = x$. We claim that either $w_1 \in sc'$ and every other $w_i \in B'$, or $w_1 \in sc''$ and every other $w_i \in B''$, from which the thesis follows. Assume $w_1 \in sc'$ (the case $w_1 \in sc''$ is symmetrical). If there exists j , $2 \leq j \leq n$, such that $w_j \in fl''$ (with $w_j \notin \tilde{\Sigma}^+$), then $sc'@fl''$ is not empty (it includes at least $w_1@w_j$), a contradiction with the hypothesis that B' and B'' are joinable. \square

Example 3. Returning to Ex. 2, we check that the two bases are joinable. The union of the bases is in decomposed form: $fl' \cup fl''$ is unproductive (because letters at positions 3, 6, ... cannot be placed); the pair (sc', sc'') is unmatchable, hence also $(sc' \cup sc'', sc' \cup sc'')$ is unmatchable. Moreover, (sc', fl'') , and (sc'', fl') are unmatchable. Therefore $L' \cup L'' = \mathcal{C}(sc' \cup sc'' \cup fl' \cup fl'')$.

For concatenation, a similar, though more involved, reasoning requires a new technical definition.

Definition 5 (Dot-product \odot and concatenability). Let B', B'' be in decomposed form, and define their *dot-product* as $B' \odot B'' = (sc' \cdot sc'') \cup fl' \cup fl''$. B' and B'' are *concatenable* if $B' \odot B''$ is in decomposed form, with scaffold $sc' \cdot sc''$ and fill $fl' \cup fl''$, and the next two clauses hold for all words $w', w'' \in \tilde{\Sigma}^+$, $y' \in sc'$, $y'' \in sc''$:

$$\exists x' \in fl' : w' = x' \cdot dot(y'') \wedge x'@y' \text{ is defined if, and only if, } w' \in fl' \wedge w'@y' \cdot y'' \text{ is defined} \quad (1)$$

$$\exists x'' \in fl'' : w'' = dot(y') \cdot x'' \wedge x''@y'' \text{ is defined if, and only if, } w'' \in fl'' \wedge w''@y' \cdot y'' \text{ is defined} \quad (2)$$

The two clauses are symmetrical. In loose terms, Clause (1) says that the fill fl' contains a word w' that matches $y'y''$, if, and only if, the word has a prefix x' , also in fl' , which matches y' , hence it is aligned with the point of concatenation. Therefore, the match $w'@y' \cdot y''$ does not produce a word that is illegal for $\mathcal{C}(B') \cdot \mathcal{C}(B'')$. This reasoning is formalized and proved next.

Theorem 2 (Concatenation of consensual languages in decomposed form). *Let the bases B', B'' be in decomposed form. If B', B'' are concatenable, then $\mathcal{C}(B') \cdot \mathcal{C}(B'') = \mathcal{C}(B' \odot B'')$.*

Proof. Let $B = B' \odot B''$.

Case $\mathcal{C}(B') \cdot \mathcal{C}(B'') \subseteq \mathcal{C}(B)$. If $x \in \mathcal{C}(B') \cdot \mathcal{C}(B'')$, then $x = x'x''$ with $x' \in \mathcal{C}(B')$, $x'' \in \mathcal{C}(B'')$. Hence, x' is the strong match of one $w' \in sc'$ (resp. $w'' \in sc''$) with $n \geq 0$ words $w'_1, \dots, w'_n \in fl' \subseteq fl$; analogously, x'' is the strong match of one $w'' \in sc''$ with $m \geq 0$ words $w''_1, \dots, w''_m \in fl''$. By definition of concatenability,

since for $1 \leq i \leq n$, every word w'_i is in fl' , then also all words $w'_1 \cdot dot(w''), w'_2 \cdot dot(w''), \dots$ are in fl' , hence also in fl . Similarly, also $dot(w'') \cdot w'_1, \dots, dot(w'') \cdot w'_n$ are in fl'' . Since $w' \cdot w''$ is in $sc'sc''$, it is possible to define a strong match yielding $x'x'' = x$, namely,

$$x = w'w'' @ (w'_1 \cdot dot(w'')) @ (w'_2 \cdot dot(w'')) @ \dots (dot(w') \cdot w'_1) @ (dot(w') \cdot w'_2) @ \dots$$

that is the concatenation of $w' @ w'_1 @ \dots @ w'_n = x'$ with $w'' @ w''_1 @ \dots @ w''_m = x''$.

Case $\mathcal{C}(B) \subseteq \mathcal{C}(B') \cdot \mathcal{C}(B'')$. Let $x \in \mathcal{C}(B)$. Then there exist $n \geq 1$ words w_1, w_2, \dots, w_n , with $n \leq |x|$, such that $w_1 @ w_2 @ \dots @ w_n = x$, $w_1 \in sc' \cdot sc''$ and $w_2, \dots, w_n \in fl' \cup fl''$. By definition, w_1 can be decomposed into $w_1 = w'_1 w'_2$ for some $w'_1 \in sc', w'_2 \in sc''$. Let $q = |w'_1|$. Assume, by contradiction, that $x \notin \mathcal{C}(B') \cdot \mathcal{C}(B'')$. Since x is the match of word $w_1 = w'_1 w'_2$ and words in $fl' \cup fl''$, the only possibility for w not being in $\mathcal{C}(B') \cdot \mathcal{C}(B'')$ is that there exists $j, 2 \leq j \leq n$, such that:

1. $w_j \in fl'$, and the substring $w_j(1, q) \notin fl'$, or
2. $w_j \in fl''$, and the substring $w_j(q+1, |x|) \notin fl''$.

We consider only Case (1) since the other is symmetrical. Since $w_j \in fl'$ and $w_j @ w'_1 w'_2$ is defined, then, by definition of concatenability, there exists $x' \in fl'$ such that $w_j = x' \cdot dot(w'_1)$, i.e., $w_j(1, q) = x'$, a contradiction with the assumption of Case (1). \square

Example 4. Consider again Ex. 2. It is easy to check that the pair $(sc' \cdot sc'', sc' \cdot sc'')$ is unmatchable, for the same reason that (sc', sc'') is unmatchable. Then, we check that the bases $sc' \cup fl'$ and $sc'' \cup fl''$ are concatenable. We only discuss the case of Clause (1) since Clause (2) is symmetrical. Let $w' \in \tilde{\Sigma}^+$, $y' \in sc', fl'' \in sc''$. If there exists $x' \in fl'$ such that $w' = x' dot(y'')$, then obviously both $w' \in fl'$ and $w' @ y' \cdot y''$ are defined.

For the converse case, assume that $w' \in fl'$ and $w' @ y' \cdot y''$ is defined. Consider the projections $\alpha = \pi_a(w')$, $\alpha' = \pi_a(y') \in (aaa)^+$ and $\alpha'' = \pi_a(y'') \in (aaa)^+$. Then $\alpha \in (aaa)^* aa(aaa)^*$. Since $w' @ y' \cdot y''$ is defined, the factor aaa of α must be matched with a factor of $\alpha' \alpha''$: by its form and alignment, the only possibility is that it is matched with a factor of α' . Hence, α has the form $(aaa)^* aa(aaa)^* dot(\alpha'')$. We omit the analogous reasoning for the projections on b . Since $w' @ y' \cdot y''$ is defined, then w' must have the form $x' \cdot dot(y'')$ for some $x' \in fl'$. Therefore $L' \cdot L'' = \mathcal{C}(sc' \cdot sc'' \cup fl' \cup fl'')$. For instance

$$\begin{aligned} a^3 b^3 a^3 b^6 &= \overset{a}{\underset{a}{\underset{a}{\text{a}}}} \overset{b}{\underset{b}{\underset{b}{\text{b}}}} \overset{a}{\underset{a}{\underset{a}{\text{a}}}} \overset{b}{\underset{b}{\underset{b}{\text{b}}}} \overset{b}{\underset{b}{\underset{b}{\text{b}}}} \overset{b}{\underset{b}{\underset{b}{\text{b}}}} @ \\ &\quad \overset{a}{\underset{a}{\underset{a}{\text{a}}}} \overset{b}{\underset{b}{\underset{b}{\text{b}}}} \overset{a}{\underset{a}{\underset{a}{\text{a}}}} \overset{b}{\underset{b}{\underset{b}{\text{b}}}} \overset{b}{\underset{b}{\underset{b}{\text{b}}}} \overset{b}{\underset{b}{\underset{b}{\text{b}}}} @ \\ &\quad \overset{a}{\underset{a}{\underset{a}{\text{a}}}} \overset{b}{\underset{b}{\underset{b}{\text{b}}}} \overset{a}{\underset{a}{\underset{a}{\text{a}}}} \overset{b}{\underset{b}{\underset{b}{\text{b}}}} \overset{b}{\underset{b}{\underset{b}{\text{b}}}} \overset{b}{\underset{b}{\underset{b}{\text{b}}}} \end{aligned}$$

This example relies on a numerical congruence with module 3 for positioning the dotted and undotted letters. We shall see how to generalize this approach to handle words of any congruence class (with respect to the length of the projections on each letter). The generalization will carry the cost of taking larger values for the congruence module.

Incidentally, we observe that the theorems of this section may have a more general use than for commutative languages. Moreover, the theorems do not require the base languages to be regular; in fact, Def. 2 applies as well to non-regular bases (as a matter of fact [3] studies context-free/sensitive bases).

3.1 A Decomposed Form Relying on Congruences

Having stated some sufficient conditions for ensuring that the union/concatenation of two consensual languages can be obtained by composing (as described by Th. 1 and Th. 2) the corresponding base

languages, we design a decomposed form, suitable for supporting joinability and concatenability, that uses module arithmetic for assigning the positions to the dotted and undotted letters within a word w over $\tilde{\Sigma}$; the preceding examples offered some intuition for the next formal developments.² Loosely speaking, each decomposed base language is “personalized” by a sort of unique pattern of dotted/undotted letters, such that, when we want to unite or concatenate two languages, the match of two words with different patterns is undefined, thus ensuring that the union or catenation of the two decomposed bases specifies the intended language composition.

For every $a \in \Sigma$, consider the projection of w on $\tilde{a} = \{a, \dot{a}\}$ and, in there, the numbered positions of each a and \dot{a} . Let m be an integer. By prescribing that for each base language, each undotted letter a may only occur in positions j characterized by a specified value of the congruence $j \bmod m$, we make the bases decomposed. We need a new definition.

Definition 6 (Slots and modules). Let $m > 3$, called *module*, be an even number. Let $R \subseteq \{1, \dots, (m/2 - 1)\}$ be a nonempty set, called a *set of slots of module m* . For every $a \in \Sigma$, define a finite language $R_m(a) \subset \tilde{a}^m$, where only positions 1 and $r + 1$ are dotted:

$$R_m(a) = \{\dot{a}a^{r-1}\dot{a}a^{m-r-1} \mid r \in R\} \quad (3)$$

The disjoint regular languages $sc\text{-}R_m, fl\text{-}R_m\tilde{\Sigma}^*$ are defined as:

$$sc\text{-}R_m = \{x \mid \forall a \in \Sigma, \pi_{\tilde{a}}(x) \in (R_m(a) \cup a)^*\} \quad (4)$$

$$fl\text{-}R_m = switch(sc\text{-}R_m) - \tilde{\Sigma}^*. \quad (5)$$

The definition of $fl\text{-}R_m$ is clearly equivalent to $\{x \mid \forall a \in \Sigma, \pi_{\tilde{a}}(x) \in (switch(R_m(a)) \cup \dot{a})^*\} - \tilde{\Sigma}^*$. It is fairly obvious that $\mathcal{C}(B) = \Sigma^+$, since $\Sigma^+ \subseteq sc\text{-}R_m$. Also, $sc\text{-}R_m @ sc\text{-}R_m = \emptyset$ and $fl\text{-}R_m$ is unproductive. The following lemma is also obvious.

Lemma 2. For all even numbers $m > 3$ and non-empty sets R of slots of module m , every base $E \subseteq sc\text{-}R_m \cup fl\text{-}R_m$ is in decomposed form, with scaffold: $E \cap sc\text{-}R_m$ and fill: $E \cap fl\text{-}R_m$.

Example 5. Let $m = 6, R = \{1, 2\}$ and $\Sigma = \{a, b\}$. Then

$$\begin{aligned} R_6(a) &= \{\dot{a}\dot{a}aaaa, \dot{a}a\dot{a}aaa\} \\ sc\text{-}R_6 &= (\dot{a}\dot{a}aaaa \cup \dot{a}a\dot{a}aaa \cup a)^* \sqcup (\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b} \cup \mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b} \cup b)^* \\ fl\text{-}R_6 &= ((\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a} \cup \mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a} \cup \dot{a})^* \sqcup (\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b} \cup \mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b} \cup \dot{b})^*) - \{\dot{a}, \dot{b}\}^* \end{aligned}$$

For clarity, in this example the characters in $sc\text{-}R_6$ and in $fl\text{-}R_6$, belonging to factors in $R_6(a), R_6(b)$, or $switch(R_6(a)), switch(R_6(b))$ respectively, are in bold. Examples of words in $\mathcal{C}(B)$ are:

$$\begin{aligned} a^6b^6 \in sc\text{-}R_6, \text{ also } a^6b^6 &= \begin{array}{c} \mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b} @ \\ \mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b} \end{array} \left| \begin{array}{l} \text{in } sc\text{-}R_6 \\ \text{in } fl\text{-}R_6 \end{array} \right. \\ \\ a^9b^8 \in sc\text{-}R_6, \text{ also } a^9b^8 &= \begin{array}{c} \mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b} @ \\ \mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b}\mathbf{b} \end{array} \left| \begin{array}{l} \text{in } sc\text{-}R_6 \\ \text{in } fl\text{-}R_6 \end{array} \right. \\ \\ (ab)^4aaabb \in sc\text{-}R_6, \text{ also } (ab)^4aaabb &= \begin{array}{c} \mathbf{a}\mathbf{b}\mathbf{a}\mathbf{b}\mathbf{a}\mathbf{b}\mathbf{a}\mathbf{b}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{b}\mathbf{b} @ \\ \mathbf{a}\mathbf{b}\mathbf{a}\mathbf{b}\mathbf{a}\mathbf{b}\mathbf{a}\mathbf{b}\mathbf{a}\mathbf{a}\mathbf{a}\mathbf{b}\mathbf{b} \end{array} \left| \begin{array}{l} \text{in } sc\text{-}R_6 \\ \text{in } fl\text{-}R_6 \end{array} \right. \end{aligned}$$

²As said, similar ideas have been used for a different language family in [4] and have been sketched for COM-SLIP languages in our communication [15].

To ensure that a base, included in $sc-R_m \cup fl-R_m$, can be used when two such languages are concatenated, we need the next simple concept.

Definition 7 (Shiftability). A language $R \subseteq \dot{\Sigma}^*$ is *shiftable* if $R = \dot{\Sigma}^* R \dot{\Sigma}^*$.

This means that any word in R remains legal, when it is padded to the left/right with any dotted words.

Next we show that by taking disjoint sets of slots over the same module, we obtain two bases that are joinable; if, in addition, the fills are shiftable, the condition for concatenability is satisfied.

Theorem 3. Let $m > 3$ and let R', R'' be two disjoint sets of slots of module m , and let $E' \subseteq sc-R'_m \cup fl-R'_m$ and $E'' \subseteq sc-R''_m \cup fl-R''_m$ be two bases. Then:

- E' and E'' are joinable;
- if the fills of E' and E'' are shiftable, then the fills of $E' \cup E''$ and $E' \odot E''$ are also shiftable, and E' and E'' are concatenable.

Proof. Let $R = R' \cup R''$. Bases E' and E'' are in decomposed form by Lm. 2. Also $E' \cup E''$ and $E' \odot E''$ are in decomposed form, since they are both subsets of $sc-R_m \cup fl-R$.

Part (1): To show that E' and E'' are joinable, we only need to prove that $(fl-R''_m, sc-R'_m)$ is unmatchable (the case $(fl-R'_m, sc-R''_m)$ being unmatchable is symmetrical). By contradiction, assume that there exist $x \in fl-R''_m$ and $y \in sc-R'_m$ such that $x@y$ is defined. Let $a \in \Sigma$ be a letter occurring in $x \notin \dot{\Sigma}^+$ and consider the projection $\alpha = \pi_{\bar{a}}(x)$. By definition of $fl-R''_m$, there exist a position q of α and a value $r \in R''$ such that $\alpha(q) = \alpha(q+r'') = a$. Then, there exists $\alpha' \in \pi_{\bar{a}}(y)$ such that $\alpha@'\alpha'$ is defined. But in α' for all positions p , $1 \leq p \leq |\alpha'|$, if $\alpha'(p) = a$ then $\alpha'(p+r') = a$ for all $r' \notin R'$. Therefore, if $p = q$ then $\alpha(p+r) = \alpha'(p+r) = a$, which is impossible by definition of matching. The same argument could be applied to show that also the other two pairs are unmatchable.

Part (2): Define as $fl-E', sc-E'$ and as $fl-E'', sc-E''$ the fills and the scaffolds of E' and E'' , respectively. If $fl-E'$ and $fl-E''$ are shiftable, then also the fill $fl-E' \cup fl-E''$ of both $E' \cup E''$ and $E' \odot E''$ is shiftable, since the union of two shiftable languages is shiftable. We now prove that in this case E', E'' are also concatenable. Let $w' \in fl-E', y' \in sc-E', y'' \in sc-E''$. If there exists $x' \in fl-E'$ such that $x'@y'$ is defined and $w' = x' \text{ dot } (y'')$, then it is obvious that $w' \in fl-E' = \dot{\Sigma}^* fl-E' \dot{\Sigma}^*$ and that $w'@(y' \cdot y'')$ is defined. We are left to show that:

$$\text{if } w'@(y' \cdot y'') \text{ is defined then } \exists x' \in fl-E' \text{ such that } w' = x' \text{ dot } (y'') \text{ and } x'@y' \text{ is defined.} \quad (6)$$

The proof of Claim (6) requires another technical definition. Given a set R of slots with module m , for $a \in \Sigma$, for every $\alpha \in \pi_{\bar{a}}(sc-R_m)$ a *restarting point* for projection α is a position i , $1 \leq i \leq |\alpha| - m$, such that $\alpha(i, i+m-1) \in R_m(a)$. Hence, at i there is a factor in $R_m(a)$. A symmetrical definition holds if $\alpha \in \pi_{\bar{a}}(fl-R_m)$: factor $\alpha(i, i+m-1) \in \text{switch}(R_m(a))$. A restarting point always exists for all $\alpha \in \pi_{\bar{a}}(sc-R_m)$ or $\alpha \in \pi_{\bar{a}}(fl-R_m)$, provided that $\alpha \notin \Sigma^+$. We claim that if $s \in sc-R_m, f \in fl-\hat{R}_m$ for some (possibly equal) sets of slots R, \hat{R} with module m , and the match $s@f$ is defined, then both the following conditions hold:

$$R \cap \hat{R} \neq \emptyset, \quad (7)$$

$$\forall a \in \Sigma, \text{ the set of restarting points for } \pi_{\bar{a}}(f) \text{ is included in the set of restarting points for } \pi_{\bar{a}}(s). \quad (8)$$

Since $f \notin \dot{\Sigma}^*$, there exists at least one $a \in \Sigma$ such that $\pi_{\bar{a}}(f)$ has a factor in $\text{switch}(\hat{R}_m(a))$ i.e., there exists a restarting point p for $\pi_{\bar{a}}(f)$. For brevity, let $\alpha = \pi_{\bar{a}}(f)$. Hence, $1 \leq p \leq |\alpha| - m$. Therefore, there exists $r \in \hat{R}$ such that $\alpha(p) = \alpha(p+r) = a$. Consider now $\beta = \pi_{\bar{a}}(s)$. Since $s@f$ was assumed to be defined,

$\beta(p) = \beta(p+r) = \hat{a}$. By definition of $sc\text{-}R_m$, $\beta \in (R_m(a) \cup a)^*$.

There are two possibilities: either p is a restarting point also for β , hence $r \in R$ and the above claims follow, or p is not a restarting point for β . The latter case is however impossible. In fact, in this case $p+r$ would be a restarting point for β , because of the form of $R_m(a)$. Therefore, since $\beta(p) = \hat{a}$, there would be a restarting point also at position $p-r'$, for some $r' \in R$. However, both r, r' , by definition, are smaller than $m/2$, therefore $2 \leq r+r' \leq m-2$. Hence, the restarting point at $p-r'$ would be at a distance less than m from the restarting point at $p+r$, which is impossible by definition of $R_m(a)$.

We prove Claim (6) to finish. For every $a \in \Sigma$, let $q'_a = |\pi_{\tilde{a}}(y')|$, and let $q''_a = |\pi_{\tilde{a}}(y'')|$. Consider the rightmost restarting point p_a for $\pi_{\tilde{a}}(w')$. By definition of $fl\text{-}E'$, there exists $r' \in R'$ such that $\pi_{\tilde{a}}(w')(p_a, p_a+m) = aa^{r'-1}aa^{m-r'-1}$. By Claim (8), p_a is also a restarting point for $\pi_{\tilde{a}}(y' \cdot y'')$: there exists $r \in R' \cup R''$ such that $\pi_{\tilde{a}}(y'y'')(p_a, p_a+m) = \hat{a}a^{r-1}\hat{a}a^{m-r-1}$. We claim that $p_a \leq q'_a$. In fact, if $p_a > q'_a$, then p_a must be a restarting point for y'' , hence $r \in R''$: but $r = r'$, a contradiction with the hypothesis that $R' \cap R'' = \emptyset$. If $p_a \leq q'_a$ then p_a must be a restarting point for $\pi_{\tilde{a}}(y')$, hence $r = r'$ and actually $p_a \leq q_a - m$. Since p_a is the rightmost restarting point, $\pi_{\tilde{a}}(w')(p_a - m + 1, q'_a + q''_a) \in \tilde{\Sigma}^+$. Choose x' to be the prefix of w' such that $w' = x' \text{dot}(y'')$. \square

4 Commutative SLIP languages and their (\cup, \cdot) -closure

This section proves the main result:

Theorem 4 (Closure under union and concatenation). *The family $\text{COM-SLIP}^{\cup, \cdot}$ is strictly included in the family of consensually regular languages: $\text{COM-SLIP}^{\cup, \cdot} \subset \text{CREG}$.*

Every language in $\text{COM-SLIP}^{\cup, \cdot}$ can be defined by an expression that combines finitely many COM-SLIP languages, using union and concatenation; since COM-SLIP is the finite union of COM-LIP languages, we may assume that the expression includes only COM-LIP, rather than COM-SLIP, languages.

In the sequel, we prove that every COM-LIP language can be consensually defined in a decomposed form such that it permits to satisfy the additional assumptions needed for union and concatenation, hence all $\text{COM-SLIP}^{\cup, \cdot}$ languages are in CREG.

Decomposed form for COM-LIP languages To expedite handling the constant terms of LIP systems, we introduce a new operation *append* that combines a language and a commutative language, the latter penetrating into the former.

Definition 8 (Appending). Let B be a language over the double alphabet $\tilde{\Sigma}$. For $a \in \Sigma$, define the (unique) factorization

$$B = B_{\tilde{a}} \cdot B_{\tilde{\Sigma}-\tilde{a}}$$

where $B_{\tilde{a}} \subseteq \tilde{\Sigma}^* \cdot \tilde{a}$ and $B_{\tilde{\Sigma}-\tilde{a}} \subseteq (\tilde{\Sigma} - \tilde{a})^*$ are languages, resp. ending by \tilde{a} , and not using the letters a, \hat{a} . If neither a nor \hat{a} occurs in B , let $B_{\tilde{a}} = \varepsilon$. Let $A \subseteq a^+$; we define the operation, named *appending A to B*, as follows:

$$B \triangleleft A = B_{\tilde{a}} \cdot (B_{\tilde{\Sigma}-\tilde{a}} \sqcup A).$$

Given a commutative language $F \subseteq \Sigma^*$, $\Sigma = \{a_1, \dots, a_k\}$, the iterative application of the previous operation to every letter of the alphabet (in any order) defines the operation, named *letter-by-letter appending F to B*, as:

$$B \triangleleft F = (\dots (B \triangleleft \pi_{a_1}(F)) \triangleleft \pi_{a_2}(F)) \dots \triangleleft \pi_{a_k}(F).$$

To illustrate, we compute:

$$\begin{aligned} \{\dot{a}\dot{b}\dot{a}\dot{b}\} \triangleleft \{ac, ca\} &= \left(\{\dot{a}\dot{b}\dot{a}\dot{b}\} \triangleleft \pi_a\{ac, ca\} \right) \triangleleft \pi_c\{ac, ca\} = \\ &= \left(\{\dot{a}\dot{b}\dot{a}\dot{b}\} \triangleleft \{a\} \right) \triangleleft \{c\} = \left(\{\dot{a}\dot{b}\dot{a}\}(\dot{b} \sqcup a) \right) \triangleleft \{c\} = \\ &= \{\dot{a}\dot{b}\dot{a}\dot{b}\dot{a}, \dot{a}\dot{b}\dot{a}\dot{a}\dot{b}\} \triangleleft \{c\} = \{\dot{a}\dot{b}\dot{a}\dot{b}\dot{a}, \dot{a}\dot{b}\dot{a}\dot{a}\dot{b}\} \sqcup \{c\} \end{aligned}$$

In the remainder of the Section, let L be a COM-LIP language over $\Sigma = \{a_1, \dots, a_k\}$, $k > 0$, defined by constant \vec{c} and periods $\mathcal{P} = \{\vec{p}^{(1)}, \dots, \vec{p}^{(q)}\}$, for some $q > 0$, with the condition that for every $\vec{p} \in \mathcal{P}$, every component p_i is even.

The next definition introduces some sets, called X, Y, W , to define the COM-LIP language L with a base D in decomposed form. The assumption on each p_i being even will be lifted when defining COM-SLIP languages.

Definition 9. For all even integers $m \geq 4$, and for all sets of slots R of the form $\{r\}$ with $0 < r < m/2$, define the regular languages $X, Y, D \subseteq \tilde{\Sigma}^*$ and the finite commutative language $W \subseteq \Sigma^*$, as follows:

$$X = \bigcup_{\vec{p} \in \mathcal{P}} \{x \in fl-R_m \mid \Psi(\pi_\Sigma(x)) = \vec{p}\} \quad (9)$$

$$Y = (R_m(a_1))^* \sqcup \dots \sqcup (R_m(a_k))^* \quad (10)$$

$$\Psi(W) = \left\{ \vec{c} + h_1 \cdot \vec{p}^{(1)} + \dots + h_q \cdot \vec{p}^{(q)} \mid 0 \leq h_1, \dots, h_q < m/2 \right\}. \quad (11)$$

$$D = X \cup (Y \triangleleft W) \quad (12)$$

It is obvious that $X \subseteq fl-R_m$. To see that $Y \triangleleft W \subseteq sc-R_m$, we first describe relevant features of the formulae. By Eq. (11), W is the finite commutative language having as Parikh image the linear subspace included between \vec{c} and $\vec{c} + (m/2 - 1)\vec{p}^{(1)} + \dots + (m/2 - 1)\vec{p}^{(q)}$. For each a_i , the projection on a_i of a word in $Y \triangleleft W$ ends with a tail of undotted a_i 's defined by Eq. (11). While the projection on a_i of $sc-R_m$ has necessarily length multiple of m , the tail does not need to comply with such constraint, thus allowing, in principle, the language $Y \triangleleft W$ to contain words whose projections on a_i has any length greater or equal to c_i (within the specified subspace). The following lemma is immediate:

Lemma 3. Let X, Y, W, D as in Def. 9. Then, D is a decomposed base included in $sc-R_m \cup fl-R_m$, with $Y \triangleleft W \subseteq sc-R_m$ being the scaffold and $X \subseteq fl-R_m$ being the fill; moreover, the fill of D is shiftable, i.e., $X = \tilde{\Sigma}^* X \tilde{\Sigma}^*$.

Example 6. Consider the language $L''_{even} = com((a^2b^4)^*)$ having the period $p_a = 2, p_b = 4$ and null constant. Notice that to obtain language $com((ab^2)^*)$, it is enough to apply union to L''_{even} and to the language $L''_{odd} = com(abb(a^2b^4)^*)$, which can be defined with the same period $p_a = 2, p_b = 4$, and with constant $c_a = 1, c_b = 2$. If module $m = 6$ and set of slots $R = \{2\}$ then $R_6(a) = \dot{a}\dot{a}\dot{a}\dot{a}^3, R_6(b) = \dot{b}\dot{b}\dot{b}\dot{b}^3$. Also, $fl-R_6 = \left((\dot{a}\dot{a}\dot{a}\dot{a}^3 \cup \dot{a})^* \sqcup (\dot{b}\dot{b}\dot{b}\dot{b}^3 \cup \dot{b})^* \right) - \{\dot{a}, \dot{b}\}^*$. Let

$$\begin{aligned} X &= \{x \in fl-R_6 \mid \Psi(\pi_{\{a,b\}}(x)) = (2, 4)\} \\ &= (\dot{a}^* \cdot \dot{a}\dot{a}\dot{a}\dot{a}^3 \cdot \dot{a}^*) \sqcup (\dot{b}^* \cdot \dot{b}\dot{b}\dot{b}\dot{b}^3 \cdot \dot{b}^* \cdot \dot{b}\dot{b}\dot{b}\dot{b}^3 \cdot \dot{b}^*) \\ Y &= (R_6(a))^* \sqcup (R_6(b))^* = (\dot{a}\dot{a}\dot{a}\dot{a}^3)^* \sqcup (\dot{b}\dot{b}\dot{b}\dot{b}^3)^* \end{aligned}$$

Both X and Y satisfy Def. 9. To complete the base of language L''_{even} , we define

$$W = \bigcup_{0 \leq i \leq 2} com(a^{2i}b^{4i})$$

The fill $\{\hat{a}, \hat{b}\}^* X \{\hat{a}, \hat{b}\}^*$ and the scaffold $Y \triangleleft W$ are a decomposed form for L''_{even} . Similarly, to define L''_{odd} , we have to define the sets X', Y', W' ; for X', Y' we select as set of slots $R' = \{1\}$, which satisfies $R' \cap R = \emptyset$. At last, $W' = \bigcup_{0 \leq i \leq 2} com(abba^{2i}b^{4i})$.

The important property of the language in Eq. (9) is stated next.

Lemma 4. 1. For all $n > 0$, for every $u \in X^{n@}$ there exist $q \geq 1$ integers $n_1, \dots, n_q \geq 0$ with $n = n_1 + \dots + n_q$ such that

$$\Psi(\pi_\Sigma(u)) = n_1 \cdot \vec{p}^{(1)} + \dots + n_q \cdot \vec{p}^{(q)}.$$

2. For all $n, n_1, \dots, n_q \geq 0$, with $n_1 + \dots + n_q = n$, if

$$u \in fl-R_m \text{ and } \Psi(\pi_\Sigma(u)) = n_1 \cdot \vec{p}^{(1)} + \dots + n_q \cdot \vec{p}^{(q)}$$

then $u \in X^{n@}$.

Proof. Part (1). By definition of X , if $x \in X$, then there exists $\vec{p}^j \in \mathcal{P}$, $1 \leq j \leq q$, such that $\Psi(\pi_{\Sigma(x)}) = \vec{p}^j$. By definition of match closure, there exists $n > 0$ words $x_1, \dots, x_n \in X$ such that $u = x_1 @ x_2 @ \dots @ x_n$. Then, for all $1 \leq i \leq n$, $\Psi(\pi_{\Sigma(x_i)}) = \vec{p}^{j_i}$ for some j_i , with $1 \leq j_i \leq q$. Hence, $\Psi(\pi_\Sigma(u)) = \sum_{1 \leq i \leq n} \Psi(\pi_{\Sigma(x_i)})$, from which the thesis follows immediately. Part (2). By definition of X , for every vector \vec{p}^j , $1 \leq j \leq q$, language X includes all words x of $fl-R_m$ such that $\Psi(\pi_\Sigma(x)) = \vec{p}^j$. Hence, one can always select n_1 words $x_1^{[1]}, \dots, x_{n_1}^{[1]} \in X$, n_2 words $x_1^{[2]}, \dots, x_{n_2}^{[2]} \in X$, etc., such that:

- i) $\Psi(\pi_\Sigma(x_i^{[j]})) = \vec{p}^j$, for every $1 \leq j \leq q$, $1 \leq i \leq n_j$;
- ii) $x_1^{[1]} @ \dots @ x_{n_1}^{[1]} @ x_1^{[2]} @ \dots @ x_{n_2}^{[2]} @ \dots @ x_1^{[q]} @ \dots @ x_{n_q}^{[q]} = u$. □

Lemma 5. The consensual language $\mathcal{C}(D)$ is commutative.

Proof. We notice first that $Y \triangleleft W$ and X obviously verify the following two conditions:

- I) $Y \triangleleft W = \pi_{\tilde{a}_1}(Y \triangleleft W) \sqcup \pi_{\tilde{a}_2}(Y \triangleleft W) \sqcup \dots \sqcup \pi_{\tilde{a}_k}(Y \triangleleft W)$;
- II) if $x \in X$ then $\pi_{\tilde{a}_1}(x) \sqcup \pi_{\tilde{a}_2}(x) \sqcup \dots \sqcup \pi_{\tilde{a}_k}(x) \subseteq X$.

Let $u \in \mathcal{C}(D)$ and let $v \in \Sigma^+$ be such that $\Psi(v) = \Psi(u)$. Word u is defined as $z @ x_1 @ \dots @ x_n$, for some $z \in Y \triangleleft W$, $n > 0$ and some $x_1, \dots, x_n \in X$. Word v is a permutation of u , hence for all $a_i \in \Sigma$ $\pi_{a_i}(u) = \pi_{a_i}(v)$. By Prop. (I) above, there exists a permutation z' of z , such that $z' \in sc-R_m \triangleleft W$, with $undot(z') = v$. Similarly, by Prop. (II) above, for all $1 \leq j \leq n$, there exists a permutation x'_j of x_j such that, for all $a_i \in \Sigma$, $\pi_{a_i}(x'_j) = \pi_{a_i}(x_j)$ and, moreover, such that $z' @ x'_i$ is defined, with $\pi_{a_i}(z' @ x'_i) = \pi_{a_i}(z @ x_i)$. Hence, also $z' @ x'_1 @ \dots @ x'_n$ is defined, therefore $z' @ x'_1 @ \dots @ x'_n = undot(z') = v$. □

Next, Th. 5 shows that D consensually defines L , with m and r arbitrarily large.

Theorem 5. For all even integers $m \geq 4$ and for every R of the form $\{r\}$, with $1 \leq r \leq m/2 - 1$, there exists a decomposed base D as in Def. 9 such that the COM-LIP language $L = \mathcal{C}(D)$.

Proof. Let m, R, D, X, Y, W be defined as in Def. 9, with $k = |\Sigma|, q = |\mathcal{P}|$. We first notice that, by definition of $Y \triangleleft W$ and of X :

(*) if $z' \in Y$ then, for every $a_i \in \Sigma$, $|z'|_{\tilde{a}_i}$ is a multiple of m , $|z'|_{\tilde{a}_i} = 2 \cdot |z|_{\tilde{a}_i}/m$ and $|z'|_{a_i} = (m-2) \cdot |z|_{\tilde{a}_i}/m$.

Proof of $\mathcal{C}(D) \subseteq L$. Let $u \in \mathcal{C}(D)$. We show that $\Psi(u) \in \Psi(L)$. Since D is in decomposed form, u must be the match of a word $z \in (Y \triangleleft W)$ with $h \geq 0$ words $x_1, \dots, x_h \in X$. Let $x = x_1 @ x_2 @ \dots @ x_h$. Word z has the form $z' \triangleleft w$ for some $z' \in Y$ and some $w \in W \subseteq \Sigma^*$. By Lm. 4, Part (1), there exist $d_1, \dots, d_q \geq 0$ such that $\Psi(\pi_\Sigma(x)) = \vec{c} + d_1 \cdot \vec{p}^{(1)} + \dots + d_q \cdot \vec{p}^{(q)}$. Also, by definition of W , there exist q integers $0 \leq h_1, \dots, h_q < m/2$ such that $\Psi(w) = \vec{c} + h_1 \cdot \vec{p}^{(1)} + \dots + h_q \cdot \vec{p}^{(q)}$. Since $u = (z' \triangleleft w) @ x$ is a strong match, $\Psi(u) = \Psi(\pi_\Sigma(z')) + \Psi(\pi_\Sigma(x)) + \Psi(\pi_\Sigma(w))$. Notice that each component of $\Psi(\pi_\Sigma(x))$ must be even: by $(z' \triangleleft w) @ x$ being a strong match it follows that $|x|_{a_i}$ is equal to $|z'|_{\tilde{a}_i}$, which is even. Again because $(z' \triangleleft w) @ x$ is a strong match, $\Psi(\pi_\Sigma(z')) = (m-2)/2 \cdot \Psi(\pi_\Sigma(x))$. Therefore:

$$\begin{aligned} \Psi(u) &= (m-2) \cdot \Psi(\pi_\Sigma(x))/2 + \Psi(\pi_\Sigma(x)) + \Psi(w) = \\ &= m \cdot \Psi(\pi_\Sigma(x)) + \Psi(w) = \\ &= m \cdot (d_1 \cdot \vec{p}^{(1)} + \dots + d_q \cdot \vec{p}^{(q)}) + \vec{c} + h_1 \cdot \vec{p}^{(1)} + \dots + h_q \cdot \vec{p}^{(q)} = \\ &= \vec{c} + (m \cdot d_1 + h_1) \cdot \vec{p}^{(1)} + \dots + (m \cdot d_q + h_q) \cdot \vec{p}^{(q)} \end{aligned}$$

Hence, $\Psi(u) \in \Psi(L)$.

Proof of $L \subseteq \mathcal{C}(D)$. For all $u \in L$ there exist q integers n_1, \dots, n_q such that $\Psi(u) = \vec{c} + n_1 \cdot \vec{p}^{(1)} + \dots + n_q \cdot \vec{p}^{(q)}$. For every j , $1 \leq j \leq q$, let $h_j = n_j \bmod (m/2)$. Let $d_j = n_j - h_j$ if $n_j p_i^{(j)} > 0$, and $d_j = 0$ otherwise. Then, every d_j and h_j are such that $0 \leq h_j < m/2$ and d_j is a (possibly zero) multiple of $m/2$. By definition of W , there exists $w \in W$ such that $\Psi(w) = h_1 \cdot \vec{p}^{(1)} + \dots + h_q \cdot \vec{p}^{(q)}$. For all $a_i \in \Sigma$, let z_i be the word in $(R_m(a_i))^*$ such that $|z_i| = d_1 p_i^{(1)} + \dots + d_q p_i^{(q)}$. Such a word does exist, since each d_j is a (possibly zero) multiple of $m/2$, hence $d_1 p_i^{(1)} + \dots + d_q p_i^{(q)}$ is a multiple of $m/2$; if this multiple is 0, then $z_i = \varepsilon$. By definition of $R_m(a_i)$, word z_i (when not empty) has, in every segment of length m belonging to $R_m(a_i)$, exactly two occurrences of \tilde{a}_i , and $(m-2)$ occurrences of a_i . Hence, $|z_i|_{\tilde{a}_i} = 2(d_1 p_i^{(1)} + \dots + d_q p_i^{(q)})/m$ and $|z_i|_{a_i} = (m-2) \cdot (d_1 p_i^{(1)} + \dots + d_q p_i^{(q)})/m$. We claim that there exists $z' \in Y$ such that $\Psi(\text{undot}(z')) = d_1 \cdot \vec{p}^{(1)} + \dots + d_q \cdot \vec{p}^{(q)}$. In fact, by Prop. (*) above, there exists $z' \in Y$ such that $\pi_{\tilde{a}_i}(z') = z_i$. Hence, $\Psi(\pi_\Sigma(z')) = (m-2) \cdot (d_1 \cdot \vec{p}^{(1)} + \dots + d_q \cdot \vec{p}^{(q)})$. By definition of W , there exists $w \in W$ such that

$$\Psi(w) = \vec{c} + h_1 \cdot \vec{p}^{(1)} + \dots + h_q \cdot \vec{p}^{(q)}.$$

Let $z'' = \text{switch}(z')$. By Lm. 4, Part (2), there exist $n = 2d_1/m + 2d_2/m + \dots + 2d_q/m$ words $x_1, \dots, x_n \in X$ such that

$$z'' = x_1 @ \dots @ x_n, \text{ with } \Psi(\pi_\Sigma(z'')) = 2 \cdot (d_1 \cdot \vec{p}^{(1)} + \dots + d_q \cdot \vec{p}^{(q)})/m.$$

Consider now $x_i \triangleleft \text{dot}(w)$. This word is in X , since the fills included in X may end with arbitrarily many \tilde{a} , for every $a \in \Sigma$. Clearly, from $x_i \triangleleft \text{dot}(w)$ one can obtain a strong match v with $z' \triangleleft w$:

$$\begin{aligned} v &= (z' \triangleleft w) @ (x_1 \triangleleft \text{dot}(w)) @ \dots @ (x_n \triangleleft \text{dot}(w)) \\ \text{with } \Psi(v) &= \Psi(\pi_\Sigma(z')) + \Psi(\pi_\Sigma(z'')) + \Psi(\pi_\Sigma(w)) = \Psi(u). \end{aligned}$$

Since the language $\mathcal{C}(D)$ is commutative, and $v \in \mathcal{C}(D)$, also $u \in \mathcal{C}(D)$. \square

We can now complete the proof of Th. 4. Since a COM-SLIP language is the finite union of COM-LIP languages, a $\text{COM-SLIP}^{\cup, \cdot}$ language is the union and concatenation of COM-LIP languages. It can be assumed that these COM-LIP languages comply with Def. 9 having only even components in every

vector of the set \mathcal{P} of periods (since otherwise they can be represented as the finite union of COM-LIP languages with this property). Select the same module and disjoint sets of slots for the decomposed bases of these COM-LIP languages. By Th. 3, since each COM-LIP is defined by a shiftable base with disjoint sets of slots, the various bases can be combined with \cup and \odot , resulting in a shiftable base. By Th. 1 and Th. 2, the result is still a consensual language (with a decomposed base). The inclusion is strict, since language $\{ba^1ba^2ba^3 \dots ba^k \mid k \geq 1\}$ has a non-SLIP commutative image, but it is in CREG [2].

5 Related Work and Conclusion

By classical results, $\text{COM-SLIP}^{\cup, \cdot}$ is included in the class of languages recognized by *reversal-bounded* multi-counter machines [1, 8] (which is also closed under concatenation). The latter class admits different, but equivalent, characterizations: as the class of languages recognized by (nondeterministic) *blind MCMs* [7], or as the minimal, intersection-closed full semi-AFL including language $\text{com}((ab)^*)$ [1, 6]. However, the cited papers are not concerned with actual construction methods for the MCMs'.

Although COM-SLIP languages have been much studied, we are not aware of any specific study on the effect on COM-SLIP of operations such as concatenation.

Concerning the techniques to specify COM-SLIP languages, our specification, using as patterns the commutative Parikh vectors, bears some similarity to Kari's [10] "scattered deletion" operation.

It is known that family COM-SLIP, when restricted to a binary alphabet, is context-free [9, 13], therefore it enjoys closure under concatenation and star. On the other hand, we observe that the intersection $I = L'^4 \cap a^+L'^2b^+$, where $L' = \text{com}((ab)^+)$, is not context-free, since

$$I \cap (a^+b^+)^4 = \{a^n b^n a^n b^n a^n b^n a^n b^n \mid n > 1\}.$$

In [13], the context-free grammar rules for COM-LIP again resemble our consensual specification.

Also, the context-sensitive grammars in [11], obtained by adding *permutative rules* of the form $AB \rightarrow BA$ to context-free grammars, include COM-SLIP and of course its closure by concatenation and star, but not its intersection with regular languages.

Last, the COM-SLIP languages are included in the SLIP language family recognized by a formal device, based on so called restarting automata, studied in [12], but the grounds covered by CREG and by that family are quite different. Beyond the mentioned similarities, we are unaware of anything related to our congruence-based decomposed form.

Unanswered questions This paper has added a piece to our knowledge of the languages included in CREG; it has introduced a novel compositional construction for the union/concatenation, which is very general and hence likely to be useful for other language subfamilies included in CREG. Some natural questions concern the closures of COM-SLIP under other basic operations: is the intersection of two COM-SLIP languages, or the Kleene star of a COM-SLIP language, in CREG?

A different kind of problem is whether the only commutative languages that are in CREG are semilinear; for instance, the nonsemilinear non-commutative language $\{ba^1ba^2ba^3 \dots ba^k \mid k \geq 1\}$ is in CREG, but, for its commutative closure, we do not know of a consensually regular specification. Last, a more general problem is whether CREG is closed under union, concatenation, and star. A possible approach is to investigate whether every CREG language may be defined by a base which is joinable and shiftable, thus obtaining closure under union and concatenation by virtue of the lemmas presented in this paper.

References

- [1] Brenda S. Baker & Ronald V. Book (1974): *Reversal-bounded multipushdown machines*. *Journal of Computer and System Sciences* 8(3), pp. 315 – 332, doi:10.1016/S0022-0000(74)80027-9.
- [2] Stefano Crespi Reghizzi & Pierluigi San Pietro (2011): *Consensual languages and matching finite-state computations*. *RAIRO - Theor. Inf. and Applic* 45(1), pp. 77–97, doi:10.1051/ita/2011012.
- [3] Stefano Crespi-Reghizzi & Pierluigi San Pietro (2012): *Strict Local Testability with Consensus Equals Regularity*. In Nelma Moreira & Rogério Reis, editors: *CIAA, Lecture Notes in Computer Science* 7381, Springer, pp. 113–124, doi:10.1007/978-3-642-31606-7_10.
- [4] Stefano Crespi Reghizzi & Pierluigi San Pietro (2013): *Deterministic Counter Machines and Parallel Matching Computations*. In Stavros Konstantinidis, editor: *Impl. and Appl. of Automata - 18th Int. Conf., CIAA 2013, Halifax, Nova Scotia, Canada, July 16-19, 2013., Lecture Notes in Computer Science* 7982, Springer, pp. 280–291, doi:10.1007/978-3-642-39274-0_25.
- [5] Seymour Ginsburgh (1966): *The mathematical theory of context-free languages*. McGraw-Hill.
- [6] Sheila A. Greibach (1976): *Remarks on the complexity of nondeterministic counter languages*. *Theor. Comp. Sc.* 1(4), pp. 269–288, doi:10.1016/0304-3975(76)90072-4.
- [7] Sheila A. Greibach (1978): *Remarks on Blind and Partially Blind One-Way Multicounter Machines*. *Theor. Comput. Sci.* 7, pp. 311–324, doi:10.1016/0304-3975(78)90020-8.
- [8] Oscar H. Ibarra (1978): *Reversal-Bounded Multicounter Machines and Their Decision Problems*. *J. ACM* 25(1), pp. 116–133, doi:10.1145/322047.322058.
- [9] Michel Latteux (1979): *Cônes rationnels commutatifs*. *J. Comput. Syst. Sci.* 18(3), pp. 307–333, doi:10.1016/0022-0000(79)90039-4.
- [10] Alexandru Mateescu (1994): *Scattered deletion and commutativity*. *Theor. Comp. Sc.* 125(2), pp. 361–371, doi:10.1016/0304-3975(94)90259-3.
- [11] Benedek Nagy (2009): *Languages Generated by Context-Free Grammars Extended by Type AB \rightarrow BA Rules*. *Journal of Automata, Languages and Combinatorics* 14(2), pp. 175–186.
- [12] Benedek Nagy & Friedrich Otto (2012): *On CD-systems of stateless deterministic R-automata with window size one*. *J. Comput. Syst. Sci.* 78(3), pp. 780–806, doi:10.1016/j.jcss.2011.12.009.
- [13] Michel Rigo (2003): *The commutative closure of a binary slip-language is context-free: a new proof*. *Discrete Appl. Math.* 131(3), pp. 665–672, doi:10.1016/S0166-218X(03)00335-4.
- [14] Arto Salomaa (1987): *Formal languages*. Academic Press, San Diego, CA, USA.
- [15] Stefano Crespi Reghizzi & Pierluigi San Pietro (2013): *Commutative consensual counter languages*. *Talk given at ICTCS 2013*, 14th Italian Conference on Theoretical Computer Science, Palermo, Italia, Sept. 9-11, 2013.
- [16] Leslie G. Valiant (1990): *A bridging model for parallel computation*. *Comm. ACM* 33(8), p. 103, doi:10.1145/79173.79181.